IFW
AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPLICANT: Phoha et al

EXAMINER: Chojnacki, Mellissa

SERIAL NO.: 10/073,453

ART UNIT: 2175

FILING DATE: February 11, 2002

DOCKET NO: 16808/95137-00

TITLE: Method for Allocation of Web Pages Using Neural Networks

ATTENTION:
Board of Patent Appeals and Interferences
Hon. Commissioner of Patents and Trademarks
Washington, D.C. 20231

TRANSMITTAL OF APPEAL BRIEF

Applicants submit herewith Appellants' Appeal Brief as well as a check in the amount of $250.00 to cover the filing fees required under 37 CFR 1.17(f).

Respectfully Submitted,

Date: June 13, 2005

BERNARD F. MERONEY
Registration No. 37,188
Jones, Walker, Waechter, Poitevent,
Carrére & Denégre, L.L.P.
5th Floor, Four United Plaza
8555 United Plaza Boulevard
Baton Rouge, LA 70809
Telephone: (225) 248-2104
Facsimile: (225) 248-3009

Attorney for Applicants

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service in an envelope addressed to: Board of Patent Appeals and Interferences, Assistant Commissioner for Patents, Washington, D.C. 20231 on this 13th day of June, 2005.

Kristen Clarke

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPLICANT: Phoha et al	**EXAMINER:** Chojnacki, Mellissa
SERIAL NO.: 10/073,453	**ART UNIT:** 2175
FILING DATE: February 11, 2002	**DOCKET NO:** 16808/95137-00

TITLE: Method for Allocation of Web Pages Using Neural Networks

ATTENTION:
Board of Patent Appeals and Interferences
Hon. Commissioner of Patents and Trademarks
Washington, D.C. 20231

CERTIFICATE OF MAILING

Date of Deposit: June 13, 2005

I hereby certify that the following attached paper or fee:

- Transmittal of Appeal Brief (1 pg.);
- Appellants' Appeal Brief - in triplicate (15 pgs.);
- Appendix of Claims on Appeal (5 pgs.);
- Check for $250.00 (appeal filing fee);
- Certificate of Mailing; and
- Stamped, return postcard;

is being deposited with the United States Postal Service on the date indicated above and is addressed to **ATTENTION: Board of Patent Appeals and Interferences**, Hon. Commissioner of Patents, Washington, D.C. 20231.

Kristine R. Crake
Kristine R. Crake

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPLICANT: Phoha et al	**EXAMINER:** Chojnacki, Mellissa
SERIAL NO.: 10/073,453	**ART UNIT:** 2175
FILING DATE: February 11, 2002	**DOCKET NO:** 16808/95137-00

TITLE: Method for Allocation of Web Pages Using Neural Networks

ATTENTION:
Board of Patent Appeals and Interferences
Hon. Commissioner of Patents and Trademarks
Washington, D.C. 20231

APPELLANTS' APPEAL BRIEF

REAL PARTY IN INTEREST
(37 C.F.R. §1.192(c)(1))

The real party in interest are the assignees, Louisiana State University, and Louisiana Tech University.

RELATED APPEALS AND INTERFERENCES
(37 C.F.R. §1.192(c)(2))

There are no related appeals or interferences.

STATUS OF THE CLAIMS
(37 C.F.R. §1.192(c)(3))

The application was filed on February 11, 2002 with fifteen (15) claims of which four (4) were independent claims (Claims 1, 10, 14 and 15).

In the First Office Action dated July 20, 2004, the examiner objected to the specification, rececated claims 11-12 on 35 USC 112 grounds, and also rejected all of the claims on 35 U.S.C. 103 grounds.

In Applicant's Amendment dated September 24, 2004, claims 11-12 were amended in response to the 112 objection.

In the Second and Final Office Action dated February 9, 2005, the Examiner maintained her rejection of all claims under 35 U.S.C. 103(b).

In Applicant's After Final Response, dated April 18, 2005, a Notice of Appeal was file, and reconsideration was requested.

In an Advisory Office Action, the request for reconsideration was denied.

The status of the claims is as follows:

allowed claims	---none
claims objected to	---none
claims rejected	---all pending

STATUS OF AMENDMENTS
(37 C.F.R. §1.192(c)(4))

There have been no amendments requested nor entered.

SUMMARY OF THE INVENTION
(37 C.F.R. §1.192(c)(5))

The invention is a method of choosing a computer, from a series of computers, to service a request for a data set I. The method is designed to select a servicing computer that has recently serviced a request for the same data set (e.g. the requested data set is in cache memory) and to avoid selecting a machine that is heavily accessed. The method uses a neural network to select the servicing machine. For a particular request for data (such as a web page), the input to the input layer of the network is a vector having vector components R(i), where R(I) is dependent upon the number of requests for the requested data set over some predetermined window. The selection of the output node to service the request (each node is associated with a particular computer) is done by choosing a path through the neural network that minimizes the distance between the vector's entry, R(i) at node I, and the neural network's weights w(I,j) at input node I. The chosen distance function can include a component for balancing service levels (load levels) across the determining system of computers as well as a component for selecting the computer(s) have recently serviced a request for the same data set. As claimed, the invention is as follows:

Claim 1.

Claim 1 is directed to method of assigning a computer to service a request for a data set within a system of a plurality of computers, where each computer has associated stored data sets. (Page 6, lines 4-15; Figure 1). The method includes the steps as follows

(1) providing a neural network with an input layer of J input nodes, an output layer having K output nodes. Each output node is associated with one of the computers in the system. The neural network has weights w(j,k) between each input node and output node.

(2) (2) receiving a request for a particular data set I and inputting to the input layer a vector having entries R(I) at input node I. R(I) is dependent upon the number of requests for the requested data over a predetermined period of time; (page 11, line 20 – page 12, line 5)

(3) selecting a computer assignment associated with a selected output node to service the data request. The selected node is associated with a specific weight that is selected to minimize a predetermined metric, where the metric measures the distance between the vector entries R(I) and the weights w(I,k) that are associated with the input node I and the output nodes k. (page 13 line 8-15)

(4) Claim 2.

Claim 2 is directed to the method of claim 1 where the method further includes the step of updating the specific weight. (page 14, lines 1-5)

Claim 3.

Claim 3 is directed to the method of claim 2 where the updating include modifying the specific weight with a factor that is dependent upon the metric distance between the vector entry R(I) and the specific weight. Page 14, lines 1-6.

Claim 4.

Claim 4 is directed to the method of claim 3 wherein the updating step includes modifying the specific weight with a means to balance the load across a subset of the output nodes. Page 14, lines 15-19

Claim 5.

Claim 5 is directed to the method of claim 4 where the means to balance is dependent upon the ratio of (number of data requests serviced by the subset of output nodes over a predetermined time period)/ (the number of output nodes in the subset). Page 14, lines 17-19.

Claim 6.

Claim 6 is directed to the method of claim 2 where the vector entry R(I) is proportional to the ratio of (the number of previous requests for the requested data)/(the number of previous requests for a subset of all data sets) over the predetermined time period. Page 13, line 22.

Claim 7.

Claim 7 is directed to the method of claim 2 where each output node is associated with a neighborhood of output nodes and the step of updating the specific weight includes updating each weight in the associated neighborhood. Page 15, lines 5-11.

Claim 8.

Claim 8 is directed to the methods of claim 2 wherein the update is provided by a formula using predetermined constants alpha, gamma and beta, as follows W(I,j)= W(I,j)+ alpha(R(I) – w(I,j)) +beta (∑W(I,k)-gamma*(W(I,j)). Page 16, line 20.

Claim 9.

Claim 9 is directed to the method of claim 1 where the input vector's components are zero, except for R(I), associated with the input node I. Page 11, lines 20-23.

Claim 10.

Claim 10 is the method of claim 1, but used in a web farm to select a server to service the data request. Page 5, line 21 through page 6, line 15.

Claim 11.

Claim 11 is directed to the method of claim 10 where the method is implemented on at least one server. Page 7 lines 4-12, claim 11.

Claim 12.

Claim 12 is directed to the method of claim 11 where the method is implemented on a least one router in the web far. Page 6, lines 13-15, claim 12.

Claim 13.

Claim 13 is directed to the method of claim 11, where the requested data is transmitted to the selected server. Page 7, lines 4-12.

Claim 14.

Claim 14 is directed to a computer readable storage medium containing executable code for performing a method of assigning a computer from a set of computers to server a request for data. The method includes the steps of association a series of weights w(I,j) with each data set I, where each j is associated with a computer; receiving a request for a data set I; associating with the requested data set a value R(I) that is dependent upon the number of requests for that data set over a predetermined period of time; and selecting a computer assignment associated with a specific weight w(I,k), where the specific weight w(I,k) minimizes the distance between the value R(I) and the weights w(I,j). Page 11 line 20, through page 13, line 15.

Claim 15 is directed to a computer readable medium having stored thereon executable instructions for performing the method specified in claim 1.

ISSUES
(37 C.F.R. §1.1912(c)(6))

The issues presented on appeal are:

Whether claims 1 – 15 are obvious under 35 USC §103(a) in view of the combination of Kakazu and Lakshmi.

GROUPING OF CLAIMS
(37 C.F.R. §1.1912(c)(7))

It is the Applicants intention that rejected claims 1, 2, 7, 14 and 15 stand or fall together; claims 3, stands alone; claims 4 and 5 stand or fall together; claims 6, 8 and 9 stand alone. Claims 10, 11, 12, 13 stand or fall together. While all claims were rejected using the same references, applicants contend that the large grouping of claims standing alone is necessitated by the additional features brought in each claim, features that simply do not appear in the references cited. Hence, based upon the art cited by the examiner, applicant believes all of these claims are independently patentable over the cited references as the references fail to teach or suggest the added features or limitations.

ARGUMENT
(37 C.F.R. §1.1912(c)(8))

The examiner has rejected claims 1-15 as obvious in light of the combination of Lakshmi and Kakazu.

THE CITED PRIOR ART

The teachings of Lakshmi:

Lakshmi is directed to methods for searching a database. Database searches can be lengthy procedures requiring extensive i/o operations, memory and processor capacity. Lakshni is directed to a technique to construct an "optimal" database search strategy, optimal in the sense of reducing costs, system resources or other desired parameters. Lakshmi uses a neural network in his technique, as follows: (1) a user search query (such as a SQL query) is received by the "optimizer"; (2) the optimizer extracts standard features of the search query to form a feature vector; (3) the feature vector is input into a neural network (NN), where the NN is trained to output or predict cost values (like expected I/O calls, selectively values for the data types, cost per call (processor resources, etc), and (4) the "cost values" are used by the optimizer to construct an optimized search strategy (for instance, to search using sequential table scans, B-tree scan indexes, etc). The optimized search strategy is provided to the database management system search engine, which then undertakes the search of the database and outputs the results of the search. See generally, Col 5, lines 22-36. The database may be located on one computer or spread across several computers.

The Examiner's Reading of Lakshmi

The examiner has indicated that Lakshmi teaches a system for receiving a data request, and assigning one computer from a plurality to service the data request. The examiner indicates that this is taught in the abstract, or in Col 33, lines 66-67, or Col 12 lines 53-63. This teaching is not present in Lakshmi nor suggested in Lakshmi. Indeed, Lakshmi does not contain such a teaching as Lakshmi is directed to techniques for optimizing database searches, that is, techniques for searching through a database. Lakshmi does not teach or suggest any means of choosing a particular computer from a plurality of computers to respond to a data request where all computers (or a subset) are capable of responding to the particular request.

Applicant's claims are not directed to "finding" data responsive to a query through a search, as does Lakshmi. Applicant's invention is directed to choosing one computer from a plurality to respond to the data request. Applicant's invention is directed to subject matter completely different than Lakshmi as Applicant does not search for data responsive to the request; instead, Applicant searches for a computer to receive the request for response, where each computer is capable of responding to the request either directly, as in cache memory, or requesting the data from a sorage device.

The examiner also indicates that Lakshmi teaches selecting a computer assignment associated with an output node of the neural network. The examiner indicates that Lakshmi teaches choosing "processing" units within the neural network (citing the abstract and column 5 lines 24-53). This area of Lakshmi simply describes any neural network which is composed of nodes or neurons (referred to in Lakshmi as "processing units"). Lakshmi does not teach "choosing" processing units within a neural network; Lakshmi simply describes feeding an input feature vector into a neural network and having the neural network process the feature vector to create the output of the network. There is no "selection" occurring here, simply feeding a feature vector as input to the neural network. The examiner is confusing the basic operation of a neural network with "choosing a computer system" through the use of a neural network. Further, no computer has been selected at the end of the Lakshmi neural network process to service the data request. Lakshmi teaches inputting a feature vector into the neural network, and the output of the neural network in not the selection of a computer, but otipmization values which will be used by the Lakshmi optimizer to generate a search query (see col. 2, lines 50-60,). The values sent from the network to the optimizer are used to construct a database query structured to optimize the search parameters. See Col 1 lines 45-65. Lakshmi does not teach selecting a computer to respond to a data request, but teaches a method of building an optimal search request to find data responsive to the search parameters. Again, Lakshmi is not directed to choosing a servicing computer, but of construction of optimal database search strategies.

Indeed, the examiner has misread Lakshmi in its entiretly. The examiner indicates that

The Teachings of Kakazu

Kakazu teaches methods to check the input/output characteristics of a neural network. Kakazu teaches checking the input/output characteristics as follows: choosing one input node, and inputting a predetermined range of variable data to the selected node while keeping constant inputs data to the remaining nodes, and examining the resultant output data from the output nodes. See col 2., lines 21-30.

The Examiner's Reading of Kakazu

The examiner has indicated that Kakazu teaches associating each output node of the neural network with a computer from a plurality of computers. The examiner indicates this teaching is in the abstract. Applicant has been unable to locate such a teaching in the abstract, nor anywhere else in Kakazu. Indeed, Kakazu does not mention multiple computers in the patent.

The examiner also indicates that Kakazu teaches inputting into the input layer of the neural network a vector R, where the entries of this vector are dependent upon the number of requests of the requested data set over a predetermined of time. The examiner cites Col.2, lines 27-30, and Col. 4, lines 62-67 for this proposition. However, a reading of theses sections indicates that an input vector is used, and one of the components of this input vector can be varied to examine the resulting the changes in the output vector. Nowhere does Kazazu teach or suggest associating the input vector components with the number of prior requests for a particular data set over a predetermined period of time. Indeed, Kakazu would not suggest such, as Kakazu does not deal with multiple computers or multiple data sets. The examiner has misread the disclosure of Kakazu.

The examiner has indicated that applicant's claim 1 does not depend upon associating input vector components with the number of prior requests. Applicant disagrees. Step (c) of claim 1 requires that the input vector has entries R(I) at each node I, where R(I) is <u>dependent upon the number of requests for the requested data over a predetermined period of time.</u>" This indicates that the input vector is dependent upon the number of prior requests that occur in a designated window (the predetermined period of time) preceeding the instant request. The examiner indicates that Kakazu teaches, in the abstract, using input vectors of previous requests; applicant has searched in vain for such language. Kakazu simply lacks this feature.

The combination of Lakshmi and Kalazu

The examiner has used the combination of Lakshmi and Kakazu on all of applicants' claims, and in particular, applicants' independent claims.

<u>In particular as to claim 1:</u> Applicants Claim 1 is method utilized in combination with a plurality

of computers, each computer having stored thereon data sets. The method uses a neural network (input nodes, output nodes and weights w(ij)) with each output node associated with a computer in the plurality of computers. Upon receiving a request for data, an input vector is formed that has an entry R(I) at input node I, where R(I) is dependent upon number of requests for the requested data over a predetermined period of time; the output of the neural network is examined to select an output node (and the associated computer) where the output node selected has a specific weight that minimizes the distance between the vector entries R(I) and the neural network weights w(I,k) associated with the input node I and the output nodes k.

Neither Lakshmi nor Kakazu alone or in combination, result in the invention in claims 1. Lakshmi nor Kalazu suggests a method to be used to chose one computer form a plurality to service a data request; neither Lakshmi nor Kakazu alone or in combination teaches or suggest using an input vector to a network having an entry that is dependent upon the number or prior requests for data (the number of requests within a predetermined window; neither Lakshmi nor Kakazu alone or in combination suggests that each output node of the neural network be associated with a computer within the plurality of computers.

The examiner is citing the Lakshmi nor Kakazu references for features that are simply not found in these references. Examples of such misreading abound, as follows:

As to claim 3;

Applicants method in claim 3 includes updating the specific weight associated with the selected output node with a factor that depends upon the distance between the entries of the input vector, and the selected weight

Neither Lakshmi nor Kakazu alone or in combination suggest such an update procedure. The examiner has cited col 5, lines 2-6, 65-67 and Col 6 lines 1-4; 65-66 of Lakshmi for this proposition. However, the examiner is incorrect. In neural network discussed in Lakshmi, must undergo a learning process during which weights are modified. However, as indicated by Lakshmi, the procedure is to modify the weights with a factor that is dependent upon the expected output and the actual output (Col 5, line 8-15). Lakshmi does not suggest or teach using the difference between the actual output and the input vector values to generate the update, as claimed in claim 3.

As to claim 4:

Applicants claim 4 requires that the update step of claim 3 for a specific weight further includes a factor to balance the load across the output nodes.

The examiner indicates that such is taught in Lakshmi col 5, lines 2-6, 65-67 and Col 6 lines 1-4; 65-66. Again, the examiner is incorrect. Nowhere does Lakshmi discuss updating a specific weight with a factor to balance loads across the output nodes. Updating only occurs in Laskami during training or learning (col 6, lins 8-10) and updating is done using factors dependent upon the expected output and the actual output (Col 5, line 8-15), not load balancing. The examiner has not explained how Laskami suggests modifying its update step to perform load balancing.

As to claim 5:

Applicants' claim 5 requires that the load balancing factor used in claim 4 is dependent upon the ratio of (number of data requests serviced by the subset of output nodes over a predetermined time period)/ (the number of output nodes in the subset).

Obviously, as Lakshmi does not discuss or suggest load balancing over the output nodes, a particular method of load balancing is not considered in Lakshmi.

As to claim 6:

Applicants claim 6 is directed to the method of claim 2 where the input vector entry value R(I) is proportional to the ratio of (the number of previous requests for the requested data)/(the number of previous requests for a subset of all data sets) over the predetermined time period.

The examiner indicates that Lakshmi teaches this in col 1, lines 25-29, col. 7 lines 24-27. The examiner is mistaken, Lakshmi, particularly the cited sections, fail to teach such limitations.

As to claims 8

Applicants' claim 8 is directed to the method of claim 2 where the update for a specific weight is given by W(I,j)= W(I,j)+ alpha(R(I) –w(I,j)) +beta (∑W(I,k)-gamma*(W(I,j)), where alpha, gamma and beta are predetermined constants.

The examiner has indicated that this is taught in Kakazu col 6, lines 39-41. However, the update formula given here is not close or related to that of applicants.

As to claim 9:

Claim 9 is directed to the method of claim 1 where the input vector's components are zero, except for R(I), associated with the input node I.

The examiner indicates that this feature is found in Lakshmi, either the abstract or the col 2, lines 47-58. However, this portion of Lakshmi describes that database management system (DBMS) characteristics as having zero or more columns of data types and zero or more user defined routines. This section has no relationship to the values of the input vector feeding the neural network.

As to claim 10.

Claim 10 is the method of claim 1, where the method is utilized in a web farm to select a server to service the data request. As noted in the specification a web farm refers to hosting web pages as data sets on more than one server. (page 6, lines 1-15) Web farms typically have a router or gateway interface between the internet and the storage data servers.

The examiner has indicated that Lakshmi discloses a Web Farm application. Again, the examiner has misread Lakshmi. Lakshmi is searching for data stored in a database. Lakshmi is not searching for a computer to service a data request. Lakshmi is not concerned with storage of multiple copies of web pages (data sets) across a web farm.

As to claim 11:

> Applicants' claim 11 is directed to the method of claim 10 where the method is implemented on at least one server.

Again, neither Lakshmi nor Kakazu are relevant to usage of the method on a server in a web farm

As to claim 12:

> Claim 12 is directed to the method of claim 11 where the method is implemented on a least one router in the web farm.

Again, neither Lakshmi nor Kakazu are relevant to usage of the method on a server in a web farm

As to claim 13:

> Claim 13 is directed to the method of claim 11, where the requested data is transmitted to the selected server.

Again, neither Lakshmi nor Kakazu are relevant to usage of the method on a server in a web farm

The combination of Lakshmi nor Kakazu simply does not result in applicants claimed invention, nor suggest that these references somehow be combined in some unknown fashion to result in applicant's invention. The examiner has filed to establish a prima facie case of obviousness as required under MPEP §706.02(j) for any of applicants' claims.

CONCLUSION

Applicants therefore believe that the application is now in condition for allowance. Applicants request that the Board of Patent Appeals and Interferences give consideration to the arguments presented herein and that the prosecution of this case be remanded to the primary examiner with a recommendation that the application be allowed.

Respectfully Submitted,

DATE: June 13, 2005

Bernard F. Meroney, Reg. #37,188
Attorney for Applicant
Jones, Walker, Waechter, Poitevent,
Carrère & Denègre, L.L.P.
4th Floor, Four United Plaza
8555 United Plaza Boulevard
Baton Rouge, Louisiana 70809
Telephone: (225) 248-2104

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service in an envelope addressed to: ATTENTION: Board of Patent Appeals and Interferences; Commissioner of Patents and Trademarks, Washington, D.C. 20231 on this 13th day of June, 2005.

APPENDIX OF CLAIMS ON APPEAL

1. (original) In a system having a plurality of computers each having data sets stored thereon, a method of assigning a computer to service a request for a data set, said method comprising the steps of:

(a) providing a neural network having at least an input layer having J input nodes and an output layer having K output nodes, each of said output nodes associated with one of said computers, and associated weights w(j,k) between each said input node and each said output node;

(b) receiving a request for particular data set I;

(c) imputing to said input layer an input vector having an entry R(I) at input node I, said entry R(I) being dependent upon the number of requests for the requested data over a predetermined period of time; and

(d) selecting a computer assignment associated with a selected one of said output nodes to service said data request, where said selected output node is associated with a specific weight, said specific weight selected to minimize a predetermined metric measuring the distance between said vector entry R(I) and the weights(I,k) associated with said input node I and said output nodes.

2. (original)The method of claim 1 where said method further includes the step of updating said specific weight.

3. (original)The method of claim 2 where said step of updating said specific weight includes modifying said specific weight with a factor dependent said metric distance between said vector entry R(I) and said specific weight.

4. (original)The method of claim 3 where said step of updating said specific weight further includes modifying said specific weight with a means to balance the load across a subset of said output nodes.

5. (original)The method of claim 4 where said means to balance the load across a subset of said output nodes is dependent upon the number of data requests serviced by said subset of said output nodes over said predetermined period of time divided by the number of output nodes in said subset of said output nodes.

6. (original)The method of claim 2 wherein R(I) is proportional to the ratio of (the number of previous requests for the requested data set) and (the number of previous requests for a subset of all data sets), over said predetermined period of time.

7. (original)The method of claim 2 wherein each output node is associated with a neighborhood of other output nodes, and said step of updating said specific weight includes updating each weight in said neighborhood of said output node associated with said specific weight.

8. (original)The method of claim 2 where said update is according to the formula $W(I,j)=W(I,j) +alpha((R(I)-w(I,j)) + beta(\Sigma W(i,k) -gama*W(I,j))$, where alpha, beta and gama are pre-determined constants.

9. (original)The method of step 1 where said input vector's components, other than said component R(I) associated with said input node I, are of value zero.

10. (amended)In a web farm of servers, a method of selecting a server to service a user request for a data set comprising the steps of:

(a) providing a neural network having at least an input layer having J input nodes and an output layer having K output nodes, each of said output nodes associated with one of said servers, and associated weights w(j,k) between each said input node and each said output node;

(b) receiving a request for particular data set I;

(c) inputting to said input layer an input vector having an entry R(I) at input node I, said entry R(I) being dependent upon the number of requests for the requested data over a predetermined period of time,

(d) selecting a server assignment associated with of one of said output nodes to service said data request, where said output node is associated with a specific weight, said specific weight selected to minimize a predetermined metric measuring the distance between said vector entry R(I) and the weights(I,k) associated with said input node I and said output nodes.

11. (amended) A method implemented in a web farm according to claim 10, where said method is implemented on at least one server in said web farm.

12. (original)A method implemented in a web farm according to claim 11 where said method is implemented on at least one router in said web farm.

13. (original)The method according to claim 1 further comprising the step of transmitting said request to said server associated with said server assignment.

14. (original)A computer readable storage medium containing computer executable code for performing a method of assigning a computer from a set of computers to service a request for a data set, said method comprising the steps of:

(a) associating for each data set I a series of weights w(I,j), where j=1,number of computers in the set of computers, associating with each individual weight w(I,j) one of said computers from said set of computers;

(b) receiving a request for particular data set I;

(c) associating with said requested data set a value R(I) being dependent upon the number of requests for the requested data set over a predetermined period of time,

(d) selecting a computer assignment associated with a specific one of said series of weights w(I,j) to service said data request, where said specific weight is selected to minimize a predetermined metric measuring the distance between said value R(I) and the weights(I,k) associated with said particular data set I.

15. (original)A computer readable storage medium containing computer executable code for performing a method of assigning a computer for a set of computers to service a request for a data set, said method comprising the steps of:

(a) providing a neural network having at least an input layer having J input nodes and an output layer having K output nodes, each of said output nodes associated with one of said computers, and associated weights w(j,k) between each said input node and each said output node;

(b) receiving a request for particular data set I;

(c) imputing to said input layer an input vector having an entry R(I) at input node I, said entry R(I) being dependent upon the number of requests for the requested data over a predetermined period of time,

(d) selecting a computer assignment associated with of one of said output nodes to revise said data request, where said output node is associated with a specific weight, said specific weight selected to minimize a predetermined metric measuring the distance between said vector entry R(I) and the weights(I,k) associated with said input node I and said output nodes.